



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/633,441	07/31/2003	Markus L. Rossmann	200206980-1	4553
22879 7590 03/18/2008 HEWLETT PACKARD COMPANY P O BOX 272400, 3404 E. HARMONY ROAD INTELLECTUAL PROPERTY ADMINISTRATION FORT COLLINS, CO 80527-2400				
EXAMINER				
FRENEL, VANEL				
ART UNIT		PAPER NUMBER		
3687				
NOTIFICATION DATE		DELIVERY MODE		
03/18/2008		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM

mkraft@hp.com

ipa.mail@hp.com

### Office Action Summary

**Application No.**

10/633,441

**Applicant(s)**

ROSSMANN ET AL.

**Examiner**

VANEL FRENEL

**Art Unit**

3627

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 7/31/08.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SF/ICE)  
Paper No(s)/Mail Date 20030731
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

**DETAILED ACTION**

Notice to Applicant

1. This communication is in response to the Application filed on 7/31/03. Claims 1-30 are pending.

***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan (2003/0093320) in view of Gryglewicz et al. (6,993,502).

(A) As per claim 1, Sullivan discloses a computer program product including program code, when executed on a computer system, for providing an interface between a calling application and at least one callable application (See Sullivan, Page 1, Paragraph 0002; Page 15, Paragraph 0129).

Sullivan does not explicitly disclose said program code representing a computer program which implements at least two controllers which cooperate with each other and are at different hierarchical levels, wherein said controllers are instances of a generic controller.

However, this feature is known in the art, as evidenced by Gryglewicz. In particular, Gryglewicz suggests that the computer program product having said program

code representing a computer program which implements at least two controllers which cooperate with each other and are at different hierarchical levels, wherein said controllers are instances of a generic controller (See Gryglewicz, Col.5, lines 55-67 to Col.6, line 30).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have included the feature of Gryglewicz within the system of Sullivan with the motivation of utilizing table-driven technologies to apply existing tax code definitions for both conventionally taxed goods and services as well as digital goods and services provided electronically over the Internet (See Gryglewicz, Col.4, lines 31-35).

(B) As per claim 2, Sullivan discloses the computer program product wherein the computer program is written in an object-oriented programming language (See Sullivan, Page 15, Paragraph 0129), and the generic controller is a class, and the at least two controllers are subclasses inherited from the generic controller class (See Sullivan, Page 15, Paragraphs 0126-0127).

(C) As per claim 3, Gryglewicz discloses the computer program product wherein the calling application is a computerized business application or an online request handling application (See Gryglewicz, Col.5, lines 46-67; Col.6, lines 61-67).

(D) As per claim 4, Gryglewicz discloses the computer program product wherein the at least one callable application is a transaction-tax service application (See Gryglewicz,

Col.7, lines 30-67).

(E) As per claim 5, Gryglewicz discloses the computer program product wherein the at least one transaction-tax service application is a transaction-tax calculation application or a transaction-tax logging application (See Gryglewicz, Col.2, lines 47-59; Col.6, lines 25-60).

(F) As per claim 6, Gryglewicz discloses the computer program product providing an interface to at least two transaction-tax service applications, said transaction-tax service applications comprising at least two different transaction-tax calculation applications (See Gryglewicz, Col.7, lines 1-50).

(G) As per claim 7, Gryglewicz discloses the computer program product wherein the controller at the higher hierarchical level is arranged for controlling an overall logic processing of the interface, and the controller or controllers at the lower hierarchical level is or are arranged for controlling a processing of the interface specific to the callable application or applications with which the respective controller is associated (See Gryglewicz, Col.8, lines 8-53).

(H) As per claim 8, Sullivan discloses the computer program product wherein the controller at the higher hierarchical level is arranged for receiving an input request from the calling application and sending an output request to the

controller at the lower hierarchical level, and receiving an output response from the controller at the lower hierarchical level as an input response (See Sullivan, Page 4, Paragraph 0047), the controller at the lower hierarchical level is arranged for receiving the output request of the controller at the higher hierarchical level as an input request, sending an output request to the callable application or one or more of the callable applications to which it is associated, receiving an input response from the callable application or applications, and sending an output response to the controller at the higher hierarchical level (See Sullivan, Page 6, Paragraph 0059).

(I) As per claim 9, Gryglewicz discloses the computer program product providing an interface to at least two callable applications of a same type, wherein the specific processing for which the controller at the lower hierarchical level and assigned to the at least two callable applications is arranged comprises deciding to which one of the at least two callable applications the output request is sent (See Gryglewicz, Col.37, lines 1-41).

(J) As per claim 10, Gryglewicz discloses the computer program product wherein the at least two callable applications of the same type are transaction-tax calculation applications (See Gryglewicz, Col.9, lines 14-41).

(K) As per claim 11, Sullivan discloses the computer program product wherein the controllers comprise at least one of the following components: an input/output module;

an input parser; a validation engine; a universal state machine; a knowledge base module; a process carrier (See Sullivan, Page 15, Paragraphs 0127-0128).

(L) As per claim 12, Sullivan discloses a computer program product including program code, when executed on a computer system (See Sullivan, Page 15, Paragraph 0129), for providing an interface between a calling application and at least one callable application, said program code including an interface architecture component, comprising: an input/output module (See Sullivan, Page 15, Paragraph 0129-0131); an input parser; a universal state machine; a knowledge base module; a process slip module; a process carrier.

(M) As per claim 13, Sullivan discloses the computer program product wherein said interface architecture component further comprises a validation engine (See Sullivan, Page 15, Paragraph 0127).

(N) As per claim 14, Sullivan discloses the computer program product wherein said interface comprises at least two cascaded interface architecture components (See Sullivan, Page 15, Paragraphs 0124- 0127).

(O) As per claim 15, Sullivan discloses a software-implemented method of interfacing a calling application and at least one callable application (See Sullivan, Page 1, Paragraph 0002; Page 15, Paragraph 0129).

Sullivan does not explicitly disclose that the software method comprising: using at least two software-implemented controllers at different hierarchical levels; performing, both with the controllers at the higher and lower hierarchical levels, a sequence of steps comprising: upon receipt of an input request from an higher hierarchical level element, which is the calling application or a controller at a higher hierarchical level, performing input request handling, sending at least one output request to at least one lower hierarchical level element, which is a controller at a lower hierarchical level or the at least one callable application, receiving an input response to the at least one output request from the lower hierarchical level element, sending an output response to the higher hierarchical level element, wherein the output request of the controller at the higher hierarchical level is the input request to the controller at the lower hierarchical level, and the output response of the controller at the lower hierarchical level is the input response to the controller at the higher hierarchical level.

However, these features are known in the art, as evidenced by Gryglewicz. In particular, Gryglewicz suggests that the software method having comprising: using at least two software-implemented controllers at different hierarchical levels (See Gryglewicz, Fig.8; Col.5, lines 45-67 to Col.6, line 60; Col.7, lines 1-44); performing, both with the controllers at the higher and lower hierarchical levels (See Gryglewicz, Col.22, lines 60-67 to Col.23, line 23), a sequence of steps comprising: upon receipt of an input request from an higher hierarchical level element, which is the calling application or a controller at a higher hierarchical level (See Gryglewicz, Fig.3; Fig.4; Col.8, lines 54-67 to Col.9, line 42), performing input request handling, sending at least



one output request to at least one lower hierarchical level element, which is a controller at a lower hierarchical level or the at least one callable application, receiving an input response to the at least one output request from the lower hierarchical level element, sending an output response to the higher hierarchical level element, wherein the output request of the controller at the higher hierarchical level is the input request to the controller at the lower hierarchical level, and the output response of the controller at the lower hierarchical level is the input response to the controller at the higher hierarchical level (See Gryglewicz, Col.10, lines 45-67 to Col.11, line 27; Col.30, lines 18-64).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have included the feature of Gryglewicz within the system of Sullivan with the motivation of utilizing table-driven technologies to apply existing tax code definitions for both conventionally taxed goods and services as well as digital goods and services provided electronically over the Internet (See Gryglewicz, Col.4, lines 31-35).

(P) As per claim 18, Sullivan discloses a method of implementing a programmed interface between a calling application and at least one callable application (See Sullivan, Page 1, Paragraph 0002; Page 15, Paragraph 0129).

Sullivan does not explicitly disclose comprising: coding at least two controllers at different hierarchical levels, wherein said controllers are instances of a generic controller (See Gryglewicz, Col.10, lines 45-67 to Col.11, line 27; Col.30, lines 18-64).

However, this feature is known in the art, as evidenced by Gryglewicz. In particular, Gryglewicz suggests comprising: coding at least two controllers at different

hierarchical levels, wherein said controllers are instances of a generic controller (See Gryglewicz, Col.10, lines 45-67 to Col.11, line 27; Col.30, lines 18-64).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have included the feature of Gryglewicz within the system of Sullivan with the motivation of utilizing table-driven technologies to apply existing tax code definitions for both conventionally taxed goods and services as well as digital goods and services provided electronically over the Internet (See Gryglewicz, Col.4, lines 31-35).

(Q) As per claim 19, Gryglewicz discloses the method wherein the controllers are coded in an object-oriented programming language, and the generic controller is a class, and the at least two controllers are subclasses inherited from the generic controller class (The examiner interprets other controllers 34<sub>I</sub>....34<sub>J</sub> to be a form of two controllers (See Gryglewicz, Col.5, lines 45-67).

(R) As per claim 27, Sullivan discloses a computer program product including program code, when executed on a computer system, for providing an interface between a calling application and at least two transaction-tax calculation applications, said interface is arranged to carry out, when called by the calling application, at least one of: selecting one of the transaction-tax calculation applications depending on a transaction attribute (See Sullivan, Page 1, Paragraphs 0002; Page 15, Paragraph 0129).

Sullivan does not explicitly disclose calling the selected transaction-tax calculation application and receiving a response from the called transaction-tax calculation application; and calling at least two of the transaction-tax calculation applications, comparing the responses returned by them.

However, these features are known in the art, as evidenced by Gryglewicz. In particular, Gryglewicz suggests calling the selected transaction-tax calculation application and receiving a response from the called transaction-tax calculation application (See Gryglewicz, Col.7, lines 1-45); and calling at least two of the transaction-tax calculation applications, comparing the responses returned by them (See Gryglewicz, Col.7, lines 1-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have included the feature of Gryglewicz within the system of Sullivan with the motivation of utilizing table-driven technologies to apply existing tax code definitions for both conventionally taxed goods and services as well as digital goods and services provided electronically over the Internet (See Gryglewicz, Col.4, lines 31-35).

(S) As per claim 28, Gryglewicz discloses the computer program product wherein the interface is further arranged to return a response to the calling application based on the response from the called transaction-tax calculation application or, if at least two transaction-tax calculation applications have been called, based on the comparison (See Gryglewicz, Col.7, lines 1-45).

(T) As per claim 29, Gryglewicz discloses the computer program product wherein the interface is further arranged to direct a response for logging purposes to a logging controller based on the response from the called transaction-tax calculation application or, if at least two transaction-tax calculation applications have been called, based on the comparison (See Gryglewicz, Col.9, lines 1-42).

(U) As per claim 30, Sullivan discloses a software-implemented method of interfacing a calling application and at least two transaction-tax calculation applications, comprising, when a call is received from the calling application, at least one of: selecting one of the transaction-tax calculation applications depending on a transaction attribute, (See Sullivan, Page 1, Paragraphs 0002; Page 15, Paragraph 0129).

Sullivan does not explicitly disclose calling the selected transaction-tax calculation application and receiving a response from the called transaction-tax calculation application; and calling at least two of the transaction-tax calculation applications and comparing the responses returned by them.

However, these features are known in the art, as evidenced by Gryglewicz. In particular, Gryglewicz suggests calling the selected transaction-tax calculation application and receiving a response from the called transaction-tax calculation application (See Gryglewicz, Col.7, lines 1-45); and calling at least two of the transaction-tax calculation applications and comparing the responses returned by them (See Gryglewicz, Col.7, lines 1-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have included the feature of Gryglewicz within the system of Sullivan with the motivation of utilizing table-driven technologies to apply existing tax code definitions for both conventionally taxed goods and services as well as digital goods and services provided electronically over the Internet (See Gryglewicz, Col.4, lines 31-35).

(V) Claims 16-17 and 20-26 recite the underlying process steps of the elements of claims 4-5 and 7-11, respectively. As the various elements of claims 4-5 and 7-11 and have been shown to be either disclosed by or obvious in view of the collective teachings of Sullivan and Gryglewicz, it is apparent that the apparatus disclosed by the applied prior art performs the recited underlying functions. As such , the limitations recited in claims 16-17 and 20-26 are rejected for the same reasons given above for claims 4-5 and 7-11, and incorporated herein.

### ***Conclusion***

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The cited but not the applied art teaches system and method for making financial updates and tracking tax status over a computer network (2004/0078307), transaction tax settlement in personal communication devices (2002/0116302), uniform data model (2003/0061061) and organizing and managing transaction-related tax information (7,043,448).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to VANEL FRENEL whose telephone number is (571)272-6769. The examiner can normally be reached on 6:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ryan Florian Zeender can be reached on 571-272-6790. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Vanel Frenel/  
Examiner, Art Unit 3627

February 29, 2008